UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/604,879	08/22/2003	Hin-Kwai Lee	NM-103	1878

23933 7590 08/11/2006

STUART T AUVINEN
429 26TH AVENUE
SANTA CRUZ, CA 95062-5319

EXAMINER	
PATEL, SHAMBHAVI K	

ART UNIT	PAPER NUMBER
2128	

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/604,879	LEE, HIN-KWAI	
	Examiner	Art Unit	
	Shambhavi Patel	2128	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

P riod for R ply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 August 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 August 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *8/22/03*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-20 are pending.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 22 August 2003 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the Examiner has considered the IDS as to the merits.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

1. **Claims 1-8 and 19-20 are rejected under 35 U.S.C. 112, second paragraph**, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The use of the term 'substantially' renders claims 1 and 19 indefinite.

All other claims are rejected by virtue of their dependency.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. **Claims 9-20 are rejected under 35 U.S.C. 101** because the claimed invention is directed to non-statutory subject matter. The Examiner asserts that the current state of the claim language is such that a

reasonable interpretation of the claims would not result in any useful, concrete or tangible product. The design is simulated as per the claim language, but no tangible result is produced.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. **Claims 1-20 are rejected under 35 U.S.C. 102(b)** as being clearly anticipated by **Smith et al. (US Patent No. 6,353,906)**, herein referred to as Smith.

Regarding claim 1:

Smith is directed to a domain-crossing verifier comprising:

a. a design scanner, receiving a textual design file specifying functions to be performed by a chip being designed, for locating domain-crossing signals generated by a first clock but sampled by a second clock, wherein the first clock and the second clock are asynchronous **(figures 3 and 4; column 3 lines 1-15)**

b. wherein the first clock has a first clock period and the second clock has a second clock period that differs from the first clock period **(figures 3 and 4)**

c. a delay randomizer that randomly selects as a random delay either a first delay value or a second delay value, the first and second delay values substantially differing by the second clock period **(column 4 lines 25-34).** *The multiplexer encapsulates the functionality of the delay randomizer* because it randomly selects the delay path of the input signal.

d. a delay applicator, coupled to the delay randomizer, for applying the random delay to a first flip-flop, the first flip-flop being clocked by the second clock but receiving one of the domain-crossing signals generated by the first clock as an input **(column 4 lines 25-34)**; *The multiplexer encapsulates the functionality of the delay applicator* because after it randomly selects the delay path of the input signal, it applies this delay to the output signal.

e. wherein the delay applicator applies a series of random delays generated by the delay randomizer to a plurality of the domain-crossing signals located by the design scanner **(column 4 lines 25-34)**

f. wherein the chip defined by the textual design file can be simulated using the random delays that differ by the second clock period to simulate logic hazards caused by domain-crossing signals **(column 4 lines 34-42)**

Regarding claim 2:

Smith is directed to the domain-crossing verifier of claim 1 wherein each domain-crossing signal passes through a synchronizer that includes the first flip-flop clocked by the second clock and generating a middle signal, and a second flip-flop that receives the middle signal and is clocked by the second clock to generate a re-synchronized signal that can be sampled by logic in a second domain clocked by the second clock **(figures 2 and 5; column 5 lines 44-67)** and wherein the synchronizers are added to the textual de-sign file by the design scanner or are already part of the textual design file **(column 4 lines 34-42)**

Regarding claims 3 and 4:

Smith is directed to the domain-crossing verifier of claim 2 further comprising a cycle simulator, performing the functions defined by the textual design file on input stimuli, the cycle simulator delaying sampling of the domain-crossing signals by the second clock by the random delays generated by the delay randomizer wherein the chip defined by the textual design file is simulated by the cycle simulator using the random de-lays that differ by the second clock period to simulate logic hazards caused by domain-crossing signals **(column 3 lines 56-65; column 4 lines 25-34, 44-63; column 5 lines 44-67)**.

Regarding claim 5:

Smith is directed to using a compiler to check for proper syntax and then generating a netlist **(column 1 lines 13-38)**. Smith discloses simulating the HDL design using Cadence, which first compiles the HDL to check for errors and then generates a final netlist.

Regarding claim 6:

Smith is directed to the domain-crossing verifier **(column 3 lines 56-65)** of claim 1 wherein the first delay value is zero and the second delay value is the second clock period, or wherein the first delay value is an arbitrary delay value and the second delay value is the arbitrary delay value added to the second clock period **(column 5 lines 56-67; column 6 lines 1-5)**.

Regarding claim 7:

Smith is directed to the domain-crossing verifier of claim 1 wherein the delay randomizer multiplies a random binary number **(column 4 lines 25-34)** by a period of the second clock to generate the random delay **(column 4 lines 59-67)**. The original signal is delayed *(randomly)* for a period of time at least equal to one clock period relative to the period of clock B *(the second clock)*.

Regarding claim 8:

Smith is directed to the domain-crossing verifier of claim 1 wherein the delay randomizer receives a seed value that specifies a starting point in a random-number sequence for generating the random delays **(column 4 lines 33-35)**.

Regarding claim 9:

Smith is directed to a method for simulating domain-crossing signals that cross from a first clock domain to a second clock domain comprising:

a. identifying a domain-crossing signal generated by a first clock in the first clock domain, but sampled by a second clock in the second clock domain **(column 5 lines 42)**

b. inserting an added delay to sampling of the domain-crossing signal by the second clock, wherein the added delay is selected from a first delay **(figure 5 delay 106)** and a second delay **(figure 5 delay 110)**, wherein the second delay is a period of the second clock greater than the first delay **(column 5 lines 16-20)** and wherein the steps of identifying a domain-crossing signal and inserting the added delay are repeated for other domain-crossing signals **(figure 6; column 6 lines 11-13)**, wherein some domain-crossing signals have the second delay selected as the added delay while other domain-crossing signals have the first delay selected as the added delay **(column 5 lines 56-67; column 6 lines 1-5)**

c. simulating a design containing the domain-crossing signals having the added delays to sampling by the second clock whereby the design is simulated with added delays on domain-crossing signals wherein the added delays differ by the period of the second clock **(column 2 lines 55-67)**

Regarding claim 10:

Smith is directed to the method of claim 9 wherein simulating the design comprises simulating from a design-language file before logic gates are synthesized, whereby domain crossing signals are verified before gate-level syntheses **(column 6 lines 7-15)**. Smith discloses the simulation of the design before gate-level implementation and also discloses implementing the method using a gate-level implementation.

Regarding claim 11:

Smith is directed to the method of claim 9, wherein simulating the design comprises simulating from a design language file to verify the domain-crossing signals before layout and wiring **(column 2 lines 55-67)**. The prior art discloses simulating only the circuit model contained in the HDL design, and does not consider the layout and wiring.

Regarding claim 12:

Smith is directed to the method of claim 9, wherein the steps of identifying the domain-crossing signals and inserting the added delay are repeated for all domain-crossing signals **(column 2 lines 55-67; column 3 lines 1-15)**. The behavioral synchronization is provided for each synchronization element in the model.

Regarding claims 13 and 14:

Smith is directed to the method of claim 12 wherein inserting an added delay comprises randomly selecting the first delay or the second delay as added delay whereby added delays are selected randomly **(column 4 lines 25-28).**

Regarding claim 15:

Smith is directed to the method of claim 12 wherein the delay randomizer multiplies a random binary number **(column 4 lines 25-34)** by a period of the second clock to generate the random delay **(column 4 lines 59-67)**. The original signal is delayed *(randomly)* for a period of time at least equal to one clock period relative to the period of clock B *(the second clock)*.

Regarding claim 16:

Smith is directed to the method of claim 12 further comprising:

a. inserting a synchronizer onto the domain-crossing signal, wherein the synchronizer comprises a first flip-flop and a second flip-flop in series **(figures 2 and 5; column 2 lines 42-67)**

b. wherein inserting an added delay comprises adding the added delay to an output of the first flip-flop before an input to the second flip-flop, whereby synchronizer delays are randomized **(column 4 lines 25-34).**

Regarding claim 17:

Smith is directed to the method of claim 12 wherein the first clock is asynchronous to the second clock **(figures 3 and 4)**.

Regarding claim 18:

Smith is directed to the method of claim 12 further comprising:

a. identifying a multi-cycle signal generated by the second clock in the second clock domain **(column 5 lines 5-10)**, wherein the multi-cycle signal is allowed more than one

period of the second clock to propagate before being sampled by the second clock **(figures 3 and 4; column 3 lines 16-25)** ;

b. inserting an added delay to sampling of the multi-cycle signal by the second clock **(column 5 lines 61-67)**, wherein the added delay is selected from the first delay and the second delay, wherein the second delay is a period of the second clock greater than the first delay (**column 6 lines 1-5**). *The multiplexer randomly selects the delay and then adds it into the system.*

c. wherein simulating the design includes simulating the multi-cycle signals having the added delays to sampling by the second clock whereby the design is simulated with added delays on multi-cycle signals wherein the added delays differ by the period of the second clock **(column 3 lines 1-25; column 5 lines 61-67; column 6 lines 1-5)**.

Regarding claim 19:

Smith is directed to a domain-crossing signal verifier comprising:

a. identifying means for identifying a domain-crossing signal generated by a first clock in a first clock domain, but sampled by a second clock in a second clock domain **(figures 2-5)**; wherein the domain-crossing signal is generated by the first clock passes through a first flip-flop and a second flip-flop in a synchronizer, the first and second flip-flops clocked by the second clock **(column 3 lines 1-15)**

b. delay randomizer means for generating a randomized delay for the first flip-flop in the synchronizer **(column 4 lines 25-42)**, wherein the randomized delay is randomly selected as either a first delay **(figure 5 delay 106)** or a second delay **(figure 5 delay 110)** wherein the second delay is substantially the first delay added to a period of the second clock **(column 5 lines 65-67; column 6 lines 1-5)**;

c. repeat means for activating the delay randomizer means to generate randomized delays for other domain-crossing signals identified by the identifying means **(column 3 lines 1-15)**

d. wherein some domain-crossing signals have the second delay selected as the randomized delay while other domain-crossing signals have the first delay selected as the randomized delay **(column 4 lines 25-42)**

e. simulating means for simulating a design containing the domain-crossing signals having the randomized delays **(column 4 lines 25-42)** for first flip-flops in synchronizers on the domain-crossing signals before sampling by the second clock **(column 5 lines 25-43),** whereby the design is simulated with randomized delays on domain-crossing signals wherein the randomized de-lays differ by the period of the second clock **(column 3 lines 1-15)**

Regarding claim 20:

Smith is directed to the domain-crossing signal verifier of claim 19 further comprising multi-cycle means, receiving a list of multi-cycle signals that are allowed more than one period of the second clock for signal propagation, for activating the delay randomizer means to generate randomized delays for a gate in a path of each of the multi-cycle signals on the list of multi-cycle signals;
wherein the randomized delays are applied to multi-cycle signals for simulation by the simulating means **(column 3 lines 1-15; column 4 lines 25-42; column 5 lines 25-43).**

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shambhavi Patel whose telephone number is (571) 272-5877. The examiner can normally be reached on Monday-Friday, 8:00 am – 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on (571) 272-2279. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

26 July 2006

KAMINI SHAH
SUPERVISORY PATENT EXAMINER

Automatic Verification of Asynchronous Circuits

TREVOR W.S. LEE
MARK R. GREENSTREET
CARL-JOHAN SEGER
University of British Columbia

Verifying asynchronous designs is difficult, since design errors may manifest themselves only under rare circumstances. This article describes how to model asynchronous designs as programs in Synchronized Transitions, a general-purpose hardware description language. The authors show how this representation facilitates rigorous, efficient verification.

SUPPOSE WE WOULD LIKE to design a transition arbiter, as suggested by Sproull et al.[1] The arbiter would have two clients, each communicating with the arbiter using three signals: R (request), G (grant), and D (done). Transition signaling should be employed. For example, if a client wanted to request the privilege and its request signal were currently low (high), the client would toggle the request signal to high (low). Each client could reside in one of the four states shown in Table 1. A correct implementation will guarantee that both clients cannot be in the privileged state at the same time.

Assume now that we have designed the circuit shown in Figure 1. Would this circuit behave as required? Traditionally, we would use simulation to verify the design. However, the unknown component delay values make this difficult. One solution is to assign random delay values to the components as they become unstable. With extensive simulation, we might hope to obtain some confidence in the correctness of the design.

Unfortunately, some errors may occur only under some particular delay distribution, making the chance of actually simulating this combination extremely small. The techniques this article describes, however, make it easy to show that the circuit in Figure 1 is *not* correct. Furthermore, the tool we describe provides the shortest sequence of transitions that lead to a state in which both clients will be in their privileged state. This information reduces the time and effort spent to find a correct design.

Synchronized transitions

Programs written in Synchronized Transitions describe both the computation and structure of digital circuits. Designers have used Synchronized Transitions for VLSI designs such as a vector-matrix multiplier[2] and a high-speed communications chip.[3] By using simple primitives based on concurrent programming, Synchronized Transitions works well for synchronous,[4] self-timed, and hybrid[3] designs. In this article, we emphasize the use of Synchronized Transitions for the design and verification of self-timed circuits. See Strunstrup[5] for a more complete description of Synchronized Transitions.

In a Synchronized Transitions program, state variables correspond to signals, while transitions model latches and combinational logic. Transitions specify state changes using multi-assignments. For example, the $\langle\langle a{=}b \rightarrow c{:=}a \rangle\rangle$ transition specifies an update of

0740-7475/95/$04.00 © 1995 IEEE

random input

10/604879

Simulation Bas d Verification of Register-Transfer Level Behavioral Synth sis T ls

R. Ernst
Institut f. Datenverarbeitungsanlagen
Technische Universitaet Braunschweig
D3300 Braunschweig, FRG*

S. Sutarwala
AT&T Bell Laboratories
Allentown, PA 18103

J.-Y. Jou
AT&T Bell Laboratories
Murray Hill, NJ 07974-2070

M. Tong
AT&T Bell Laboratories
Murray Hill, NJ 07974-2070

Abstract

We present a simulation based system for verification of register-transfer level behavioral synthesis tools. Applications are tool debugging and automatic regression test. Key feature is a transformation of sequential circuits for application of pseudo-random test patterns. The results show a high relevance of verification with pseudo-random patterns.

Introduction

Part of the CAD-work in the Bell Laboratories is the development of register-transfer level behavioral synthesis tools. Reliable correctness of the software is crucial as more and more designers use behavioral circuit descriptions as design entry level. We were looking for a tool for *debugging* and *regression test* of synthesis systems. The tool to be verified was BESTMAP [1] . BESTMAP accepts a register-transfer level description of a circuit in a subset of the language "C" as input and creates a gate structure as output which is implemented with standard cells.

The first step is to check, whether formal automatic verification is feasible. Automatic formal verification of gate level circuit descriptions became popular with the advent of efficient algorithms for symbolic manipulation of binary decision diagrams[2] (improvement through efficient variable ordering see[3] [4]). Those algorithms are only effective, if the functions are not "black boxes", but can be broken up into small subfunctions, which are individually analyzed before the results are composed.

Unfortunately, the behavioral model is such a "black box". A possible way would be to map the behavioral description to a gate level reference structure and then to verify the equivalence of synthesized and reference structure. In [5] such an approach is used to verify the equivalence of "hardware flowcharts" and gate level descriptions. In our case, however, the mapping from behavior to circuit structure itself is a major task of the synthesis system including C-model parsing, word length calculation, don't care condition detection and multiple assignment resolution. Similar mapping algorithms would be used for verification and synthesis, and, consequently, we cannot expect that the verification software is significantly more reliable than the synthesis tool.

Other verification approaches use symbolic manipulation for a direct comparison of hardware descriptions at different levels of abstraction, in some cases supported by exhaustive simulation ("hybrid simulation", see[6]; this paper also contains a brief overview of verification systems). Similar to the first approach, it would be necessary to partition the behavioral description in subfunctions requiring a C-model analysis which leads to comparable problems as mentioned above.

It seems that, at the current state, there is no practical way of automatic formal verification. We decided, therefore, to choose a simulation approach using a standard in-house mixed level simulator.

We present the simulation based verification system *TSG* (Test System Generator), which has been developed for the verification of register-transfer level synthesis tools. The system provides an automatic functional comparison of the input description and the output structure.

In the next section, we will take a short look at the synthesis system BESTMAP. Then, the verification system TSG is presented. The system applies random stimuli patterns for verification. In a further section, we will see how both behavioral description and synthesized structure are transformed to improve the effectiveness of random patterns. Finally, experiences using TSG in tool development are summarized.

The Synthesis System BESTMAP

As input, BESTMAP takes a register-transfer level behavioral circuit descriptions in a subset of the programming language "C". Without modification, the circuit description can be used as circuit model for our in-house simulation tools [7] [8] .

There are currently two major limitations for the C-description:

- all states (registers) in the model must be controlled by the same clock. The states may be loaded asynchronously.

* The work was done while the author was with AT&T Bell Laboratories in Allentown, PA 18103